Algebraic path tracking to aid the manual harvesting of olives using an automated service unit

Fernando A. Auat Cheein[1*], Gustavo Scaglia[2], Miguel Torres-Torriti[3], José Guivant[4], Alvaro Javier Prado[1], Jaume Arnò[5`], Alexandre Escolà[5], Joan R Rosell-Polo[5]

[1]Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile

[2]Instituto de Ingeniería Química, National University of San Juan, San Juan, Argentina

[3]Pontificia Universidad Católica de Chile, Santiago, Chile.

[4]School of Mechanical Engineering, University of New South Wales, Australia

[5]Research Group on AgroICT & Precision Agriculture, Department of Agricultural and Forest Engineering, Universitat de Lleida – Agrotecnio Center, Lleida, Spain

[*] Fernando Auat Cheein, fernando.auat@usm.cl

Av. España 1680, Valparaíso, Valparaíso, Chile

Phone: +56 322652617

Abstract:

Service units used in precision agriculture are able to improve processes such as harvesting, sowing, agrochemical application, and manure spreading. This two-part work presents, a path tracking controller based on an algebraic approach for an articulated service unit, suitable for embedded applications, and its implementation to a hierarchical navigation strategy to aid a manual harvesting process. The path tracking controller approach can be scaled to several trailers attached to the service unit. For harvesting, the service unit drives within an olive grove environment following the previously developed path and a trailer is used as a mobile hopper where olives, collected by human labour, are deposited. The service unit also registers and geo-references the amount of olives (mass) collected for the subsequent creation of yield maps. The developed navigation strategy improved the time associated with harvesting olives by approximately 42 - 45%. The mathematical formulation of the problem, some real time experimental results, the creation of a yield map and the statistical analysis that validated the method are included.

Keywords: Robot control, service robot, robot programming, agricultural engineering, harvesting aid, yield mapping, robot traction

1. Introduction

Over recent years, agriculture in Chile has experienced serious challenges affecting its productivity including low water resources, loss of agricultural land due volcanic eruptions and earthquakes. The mining industry has also offered challenges by offering better salaries and more stable jobs, causing agricultural workers to migrate to mining areas. Thus, to improve

agricultural productivity and competitiveness, it has become necessary to introduce and develop agricultural automation; more specifically to seek to introduce robotic machinery (i.e. automated service units) for both primary (harvesting, seeding, fertilising, spraying) and secondary tasks (grove supervision, weed detection, hauling, mowing). However, there are several unsolved issues that must be considered in the implementation of automated service units, (named as *co-robot* by the USA National Science Foundation), such as the interaction of the units with field workers, manoeuvring problems when operating in groves (considered as unstructured and constrained environments), task scheduling, accurate and robust environment understanding (localisation, mapping and scene classification), identification of terrain regions for safe traversing, and efficient manoeuvring and slippage handling. These problems directly affect agricultural productivity and they have still not been fully addressed despite recent research.

With the continuous increase in world population and the stresses that this places on food production, agricultural tasks are being modernised in order to increase agricultural production despite the decreasing supply of field workers found in some countries (Noguchi et al. (2004)). Over the last two decades, the robots have shown to be efficient tools for improving agricultural tasks developing what is known as precision agriculture (Lee et al. (1999); Perez-Ruiz et al. (2011); Malinowski & Yu (2011), Wood et al. (2014); Matsuda et al. (2014)).

Harvesting, sowing and agrochemical application have been successfully implemented on mobile robots -called *service units* (Auat Cheein & Carelli (2013)) within agricultural environments (Cariou et al. (2009)). For example, in Pota et al. 2007, the authors presented a robotic tractor for operation in precision agriculture for tasks such as pesticide application or crop scouting. The tractor contained a differential global navigation satellite system (GNSS) receiver, vision systems and an on-board computer for information processing and control. In

64 Rath & Kawollek (2009), a robotic harvesting mechanism aided by an artificial vision system for
65 cutting flower pedicels was presented. Tanigaki et al (2008) proposed a cherry-harvesting robot
66 that automatically collects cherries but does not navigate autonomously along the orchard alleys.
67 The system was aided by an infrared vision sensor to detect the cherries. Murakami et al. (2008)
68 presented a tele-operated manure spreader for planar terrains; the vehicle incorporated an
69 omnidirectional vision system to aid the tele-operation process but in this application, the motion
70 of the vehicle was governed by an operator. An automatic guidance system for white asparagus
71 harvesting was developed by Dong et al. (2011),. The system was based on ultrasonic sensors
72 located on each wheel of the vehicle, although the movements of the vehicle remain remotely
73 controlled by an operator. Bochtis et al. (2015) developed a comprehensive analysis and
74 procedure for routing in orchards and Anjom et al. (2014) proposed a service unit for strawberry
75 harvesting.

76 The motion control of service units is associated with the navigation strategy adopted for
77 them (Park & Lee (2011); Bochtis et al. (2010); Mas et al. (2008)). This was shown in Perex et
78 al. (2008), where a multi-agent fuzzy-based system was presented for the navigation of service
79 units in agricultural environments. A multi-vehicle control strategy based on artificial
80 intelligence techniques was presented Razaee & Abdollahi (2014); the mobile units were part of
81 a flexible manufacturing system. Feng & Zhang (2011) developed a multi-robot trajectory that
82 can be used for general agricultural purposes, although kinematics restrictions were not taken
83 into account in trajectory generation.

84 Tanaka & Murakami (2009) proposed a line tracking control for a two-wheeled bicycle
85 vehicle based on classical control theory. Huan & Tsai (2009) proposed an embedded controller
86 for omnidirectional robots. Although the controller was successfully embedded on an field

87 programmable gate array (FPGA), the mobile robot used did not consider the kinematic
88 constrains found in bicycles or a car-like vehicles, which makes it unsuitable for agricultural
89 applications.

90 A behaviour-based architecture for the navigation of mobile robots was proposed by Lian
91 (2011). Hwang & Shih (2009) used an active vision system and took into account the vehicle
92 kinematic restrictions for the navigation of car-like robots. A trajectory controller based on
93 switching graphs for a team of robots was also proposed. The implementation of such controllers
94 requires full knowledge of the position of each robot. The Lyapunov theory for control tracking
95 can also be used, as shown by Martins et al. (2011) and Blazic (2011).

96 Despite the fact that a wide variety for controllers is available in the scientific literature, most
97 of them (such as Lyapunov-based, graph-based or adaptive controllers) are difficult to implement
98 on embedded systems (Park et al. 2010; Coulaud et al 2006). Furthermore, approximations of
99 hyperbolic functions, such as commonly used in Lyapunov-based controllers (Blazic, 2011) or
100 more complex functions, are difficult to implement in embedded systems due to approximation
101 errors that can cause instability of the controller. Such limitations become crucial when working
102 in agricultural environments (Auat Cheein & Carelli (2013); Auat Cheein & Scaglia (2014)). It
103 should also be noted that, despite the efforts of the scientific and technical community to
104 automate service units, there still remain open issues regarding their functionality and their
105 contribution to the improvement of agricultural production (Auat Cheein & Carelli (2013)).

106 In this context, an asymptotically stable path tracking controller, based on an algebraic
107 approach, is presented for use in an articulated robot. The algebraic formulation of the controller
108 makes it suitable for embedded applications. The articulated service unit used in this work was
109 an unmanned vehicle with a passive trailer attached to it. Therefore, the control commands were

only generated for the service unit. The service unit incorporated a laser range sensor, odometry sensors and an real time kinematics (RTK) and a GNSS receiver that estimated the pose -position and orientation- of the unmanned vehicle. The trailer had a load cell help to build yield maps of the olive trees plot and provide a heading sensor. In addition, a precision agriculture case study is presented. It is based on the implementation of the path tracking controller for an olive harvesting aid for field workers. In order to do so, a navigation strategy was implemented. The navigation strategy can be divided into three main stages: harvesting, the full load situation and emergency stops. The first stage was associated with the harvesting process: the concept was that as the service unit navigated along the alleys of the olive grove, the workers deposit harvested olives in the trailer. When the trailer detected a full load situation, it re-planned the path in order to reach the shelter as soon as possible where it unloaded the olives. Once the trailer was empty, the service unit returned to the olive field to continue the harvesting aid. If a field worker was detected as being close to the service unit during its navigation, the vehicle stopped its motion in order to protect the worker's integrity and its own. Once the path was clear again, the service unit continued its navigation. Several real time experiments were included in order to show the performance of the concept. Also, as the trailer was equipped with a load cell, weight data was acquired and geo-referenced in order to map the mass spatial distribution of yield in the grove. Such information is essential to implement precision agriculture strategies as stated in Alamo & Feito (2012), who concluded that detailed analysis of harvest can improve crop management and lead to input reductions and environmental benefits. The yield maps from manually harvested olive groves are usually created by manually weighing and/or geo-referencing the sacks, bins or boxes used. An automated weighing and geo-referencing for the yield can improve the creation and the reliability of yield maps.

133 This article is organised as follows: in the section 2, the vehicle used in this work, its
134 kinematics constrains and technological aspects; the mathematical formulation of the path
135 tracking controller proposed in this work; the precision agriculture case study: i) the path
136 tracking controller implemented on the service unit with the attached trailer for olive harvesting
137 tasks; and ii) the mass of harvested olives registered at each trailer geo-referenced location for
138 further yield map construction is described. The results obtained during the experiments
139 regarding service unit navigation, harvesting times and yield mapping are described in section 3.
140 In section 4 the lessons learned during the work are discussed and conclusions drawn are
141 presented in section 5.
142
143 **2. Materials and Methods**
144 *2.1 Service Unit Model*
145 The proposed path tracking controller and navigation strategies were tested using an
146 autonomous vehicle (a robotised quad-cycle) used for the supervision of olive groves, as shown
147 in Fig. 1. Such a robotic service unit is endowed with turn encoders on all wheels and an RTK-
148 GNSS (as used in Perez-Ruiz et al. (2012)) that is employed as the main source for position
149 measurements. For the purpose of the experiments carried out here, a passive trailer (shown in
150 the middle picture in Fig. 1) was attached to the rear of the vehicle. The trailer had a weight
151 sensor that was employed to estimate the amount of harvested olives. It is important to mention
152 that the service unit is able to tow a trailer with a maximum payload of 300 kg. When fully
153 loaded, the power transmission allows the robot to reach a maximum longitudinal velocity
154 $v^x = \pm 3$ m s^{-1} a maximum heading $\psi = \pm 45°$.

Fig. 1: Service unit used in this work, its passive trailer and its corresponding kinematic model. Left shows a bicycle-like robot, middle is the trailer used during the harvesting experiments, right shows a schematic of the kinematic model of the service unit with a trailer.

The service unit was equipped with a guidance computer based on a standard industrial PC with an Intel Core 2 Duo L7500 1.6 GHz processor and 2 GB RAM of memory. The additional guidance computer accepted longitudinal velocity and heading commands from the path tracking controller. The general system architecture of the system is shown in Fig. 2. As can be seen, the two computers were networked, one for processing low level information (encoders' readings, inertial module unit's readings, and pay-load reading) and a second computer for processing exteroceptive information, such as the LiDAR readings or calculating control actions. In this work, the LiDAR was only used for acquiring exteroceptive information from the environment, as will be shown in section 3.

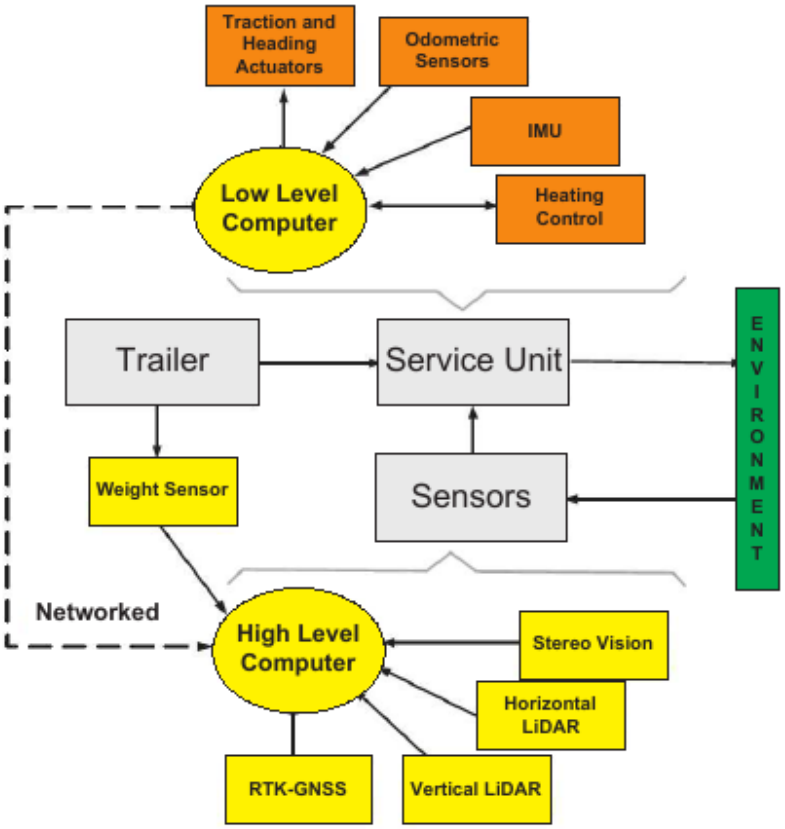


Fig. 2: General architecture of the service unit. The service unit computational design consists of two networked computers which process both exteroceptive (associated with the GNSS, LiDARs and artificial vision systems) and proprioceptive (associated with the low control level, IMUs and weight sensor) information.

For the purpose of implementing the path tracking controller, the motion of the service unit can be described in terms of the standard kinematic model of a car-like drive mobile platform, which is given by:

$$\begin{pmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{pmatrix} = \begin{pmatrix} v_t^x \cos(\theta_t) \\ v_t^x \sin(\theta_t) \\ \frac{v}{L} \tan(\psi_t) \end{pmatrix} \quad (1)$$

where $(x_t, y_t, \theta_t,)$ are the pose of the robot in the world reference frame, v_t^x is the longitudinal velocity input, ψ_t is the heading input and L is the distance between axles (see Fig. 1).

However, since the service unit has a passive trailer attached to it, the complete kinematic model for the vehicle with a trailer being pulled at an angle $\psi_{2,t}$ relative to the longitudinal-axis of the traction vehicle is given by:

$$\begin{pmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \\ \dot{\psi}_{2,t} \end{pmatrix} = \begin{pmatrix} v_k \cos(\psi_{1,k}) \cos(\theta_t) \\ v_k \cos(\psi_{1,k}) \sin(\theta_t) \\ \frac{v_t}{L_1} \sin(\psi_{1,k}) \\ -\frac{v_t \cos(\psi_{1,t})}{L_2} \sin(\psi_{2,t}) \end{pmatrix} \tag{2}$$

where L_1 is the distance between axles of the traction vehicle and L_2 is the tow bar length between the trailer and the tow ball joint, which for simplicity has been assumed to be located on the rear-axle of the traction vehicle, as shown in the layout of Fig. 1. In this work, for the vehicle shown in Fig. 1, $L_1 = 1.5\text{m}$, $L_2 = 1.5\text{m}$ and the maximum value for ψ_1 is $\pm 45°$. These values and the trailer itself result in a kinematic constraint which imposes a minimum turning radius $r_{minimum} = 3m$.

2.2 Path Tracking Controller Design

The main objective of the path tracking controller is to drive the vehicle with a trailer attached to it. The trailer is passive, that is, no control actions are imparted to it and it follows the unmanned vehicle to which it is attached as it navigates through the environment. The path tracking controller approach follows the guidelines previously published in Auat Cheein & Scaglia (2014) for algebraic-based controller design. In addition, in relation to the objectives of

this research, no significant rotations (i.e. roll, pitch and yaw) and slippage were expected to occur in the motion of the vehicle, as stated in Wang et al. (2011).

$$x_{k+1} = x_k + v_k \cos(\psi_{1,k})\Delta_t$$
$$y_{k+1} = y_k + v_k \sin(\psi_{1,k})\Delta_t \quad (3)$$
$$\theta_{k+1} = \theta_k + \frac{v_k \sin(\psi_{1,k})}{L_1}\Delta_t$$

Equation (3) shows the discrete kinematic model of the service unit; the suffix *k* represents the discrete time and ψ_1 is the steering wheel (heading) angle relative to the longitudinal axis of the vehicle, and *v* is the vehicle's longitudinal speed, as shown in Eq. (2). Since the trailer is passive, its movement was not included in Eq. 3. The objective is to calculate v_k and $\psi_{1,k}$, the control signals of the car-like service unit, in order to drive the vehicle according to a previously defined trajectory reference: $\Omega = [x_{ref}, y_{ref}]^T$. Such trajectory reference Ω must be kinematically compatible with the vehicle kinematics restrictions. From Eq. (3), and by re-arranging its terms, it is possible to see that, the first two expressions of Eq. (3) form an equation system with a single unknown of the form shown in Eq. (4).

$$\begin{bmatrix}\cos(\theta_k)\\ \mathrm{sn}(\theta_k)\end{bmatrix} v_k \cos(\psi_{1,k}) = \frac{1}{\Delta_t}\begin{bmatrix}e_{x,k}\\ e_{y,k}\end{bmatrix} + \begin{bmatrix}x_{k+1} - x_k\\ y_{k+1} - y_k\end{bmatrix} \quad (4)$$

Considering that Ω is beforehand known, then it is possible to replace x_{k+1} and y_{k+1} by $x_{ref,k+1} - k_x(x_{ref,k} - x_k)$ and $y_{ref,k+1} - k_y(y_{ref,k} - y_k)$ respectively, with $0 < k_x, k_y <$

1 to ensure the zero error convergence of the trajectory follower -i.e., $||e_k||_2 = \sqrt{e^2{}_{x,k} + e^2{}_{y,k}} \to 0$ as $k \to \infty$ where $||.||_2$ is the 2-norm with $e_{x,k} = x_{ref,k} - x_k$ and $e_{y,k} = y_{ref,k} - y_k$-. Then,

$$\begin{bmatrix}\cos(\theta_k)\\ \mathrm{sn}(\theta_k)\end{bmatrix} v_k \cos(\psi_{1,k}) = \frac{1}{\Delta_t}\begin{bmatrix}\Delta_x\\ \Delta_y\end{bmatrix} = \frac{1}{\Delta_t}\begin{bmatrix}x_{ref,k+1} - k_x(x_{ref,k} - x_k) - x_k\\ y_{ref,k+1} - k_y(y_{ref,k} - y_k) - y_k\end{bmatrix} \quad (5)$$

Equation (5) has exact solution when the following condition is achieved (Auat & Scaglia (2014)).

$$\tan((\theta_k) = \frac{\sin(\theta_k)}{cos(\theta_k)} = \frac{y_{ref,k+1} - k_y(y_{ref,k} - y_k) - y_k}{x_{ref,k+1} - k_x(x_{ref,k} - x_k) - x_k} = \frac{\Delta_y}{\Delta_x} \quad (6)$$

In Eq. (6), the angle θ_k that accomplishes the condition, will be named as θ_{ez}. From Eq. (3), $\theta_{k+1} = \theta_k + \frac{v_k \sin(\psi_{1,k})}{L_1}\Delta t$. Replacing θ_{k+1} by $\theta_{ez,k+1} - k_\theta(\theta_{ez,k} - \theta_k)$ yields:

$$v_k \sin(\psi_{1,k}) = \frac{\theta_{ez,k+1} - k_\theta(\theta_{ez,k} - \theta_k) - \theta_k}{\frac{\Delta_t}{L_1}} \quad (7)$$

In addition, according to Eq. (5) and Eq. (6).

$$v_k \cos(\psi_{1,k}) = \frac{\Delta_x}{\Delta_t}\cos(\theta_{ez,k}) + \frac{\Delta_y}{\Delta_t}\sin(\theta_{ez,k}) \quad (8)$$

and re-arranging the terms of Eqs. (7) and (8) the following equation system with a single unknown is obtained.

$$\begin{bmatrix} v_k \sin(\psi_{1,k}) \\ v_k \cos(\psi_{1,k}) \end{bmatrix} = \frac{1}{\Delta_t} \begin{bmatrix} \Delta_x \cos(\theta_{ez,k}) + \Delta_y \sin(\theta_{ez,k}) \\ (\theta_{ez,k+1} - k_\theta(\theta_{ez,k} - \theta_k) - \theta_k) L_1 \end{bmatrix} \quad (9)$$

230

231 As previously stated for Eq. (5), Eq. (9) will have an exact solution when the following
232 condition is achieved,

$$\tan(\psi_{1,k}) = \frac{\sin(\psi_{1,k})}{\cos(\psi_{1,k})} = \frac{(\theta_{ez,k+1} - k_\theta(\theta_{ez,k} - \theta_k) - \theta_k) L_1}{\Delta_x \cos(\theta_{ez,k}) + \Delta_y \sin(\theta_{ez,k})} \quad (10)$$

233

234 Therefore,

$$v_k = \left(\frac{\Delta_x}{\Delta_t} \cos(\theta_{ez,k}) + \frac{\Delta_y}{\Delta_t} \sin(\theta_{ez,k})\right) \cos(\psi_{1,k}) + \quad (11)$$

$$+ (\theta_{ez,k} - k_\theta(\theta_{ez,k} - \theta_k) - \theta_k) \frac{L_1}{\Delta_t} \sin(\psi_{1,k})$$

235 In Eq. (11), $\psi_{1,k}$ was obtained by applying the inverse tangent to the expression shown in
236 Eq. (10). Thus, Eq. (10),its inverse tangent, and Eq. (11) are the expressions of the path tracking
237 controller for the unmanned service unit used in this work. The convergence proof of the
238 controller is included in the appendix to this paper.

239

240 *2.3 Implementation of the Path Tracking Controller Design*

241 This section shows the implementation of the service unit, its trailer and the path tracking
242 controller presented in the previous section, to a precision agriculture approach associated with
243 olive harvesting. The basic scenario is as follows:

244 • The agricultural environment is an experimental olive grove.
245 • The unmanned service unit and its trailer navigate autonomously along the alleys of the
246 olive grove.

- As the service unit navigates, its trailer is loaded with olives collected from the trees, in three selected locations spread out along each alley.
- When the weight sensor of the trailer measures a full load situation (set to 300 kg), the vehicle then drives to the shelter where the olives are unloaded.
- Once the trailer of the service unit is empty, the mobile system returns to the olive grove and continues the harvesting process, from the point in the environment where it remains.
- The path followed by the vehicle for harvesting or unloading olives, is a previously defined path.

Each step of the application will be explained as follows. Figure 3 (left) shows a satellite image of the olive grove where the experimentation was carried out. Figure (3) (right) shows the olive trees and the alleys of the environment and the solid red dot in Fig. (3) (left) shows the shelter where the harvested olives are unloaded. The solid blue line represents the mandatory path that the unmanned vehicle should follow in order to enter the olive grove. An irrigation canal bounds part of the grove, as it is shown in solid green line in Fig. (3). Manoeuvres near the canal are considered dangerous for the integrity of the vehicle and should be avoided.

Fig. 3: Olive grove environment. Left shows a satellite image of the olive grove environment where the experimentation was carried out; the solid red dot marks the shelter where the olives are stored; the solid blue line is the mandatory path that the vehicle should follow to enter into the grove; the solid green line represents the grove irrigation canal. Right shows a picture of the grove with weeds removed.

The experimental olive grove environment shown in Fig. (3) has 26 alleys of 6 m wide and 80 m long (1.25 ha). The distance between olive trees from a similar row is 2.5 m. Each olive tree produces approximately 45 kg of olives. The olive trees were 15 years old.

2.3.1. Navigation System Architecture

The navigation architecture used in this study is shown in Fig. (4). It can be summarised as follows:

- The trajectory generation process used the environment and the vehicle pose information to manage the trajectory to be followed by the service unit. The trajectory (Ω) was generated beforehand.
- The estimation of the pose of the vehicle within the olive grove was based on an RTK-GNSS, as the one used in Perez-Ruiz et al. (2012).
- Vehicle pose information was compared with the planned path. Comparison errors were then fed into the path tracking controller presented in Section *Path Tracking Controller Design*, Eqs. (10) and (11).
- In addition, the vehicle incorporated a laser range sensor that was used for collision avoidance purposes. Thus, if an obstacle, e.g. a field worker, was near to the robot, then the vehicle stopped its motion until the obstacle was out of its range. In this work, the safety range was set to 2 m. In Fig. 4, the *obstacle alert* block managed the

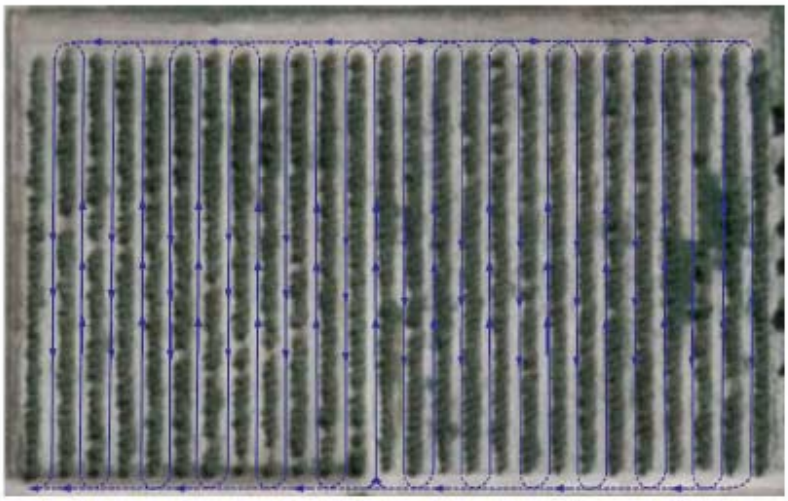
obstacle detection procedure. If an obstacle was detected, then the *obstacle alert* block had the maximum priority and interrupted the trajectory control motion until the path was free for navigation.

- The *full load alert* block indicated when the vehicle had to release its olive load. Thus, using the previously planned path, the system re-planned its trajectory in order to reach the shelter to release the collected olives, and then come back to continue acting as a harvesting aid. This situation is explained in detail in section 2.3.2. It is worth noting that, as the vehicle moved to/from the shelter, the *obstacle alert* was still operating.

2.3.2. Path Management

The path management strategy for the olive grove environment shown in Fig. 3 is based on a curly path generated along alleys based on previously available GNSS coordinates of the olive grove, similar to that shown in Bochtis et al. (2010). Due to the fact that the service unit had a trailer attached to it, the path management avoided the generation of paths that could cause a reverse movement of the vehicle. Therefore, no backward motion was considered (for the safety of the vehicle). Figure 5 shows the general trajectory followed by the service unit during its harvesting aid task. In Fig. 5, the blue line is the path established for the harvesting process; the service unit and its trailer must follow such a path. Although in this work, the path shown in Fig. 5 was used, other paths could also be applied since path planning is not the focus of this research. The solid blue segments correspond to the portion of the path in which the service unit has to perform the harvesting task. The dotted blue line represents the path where the vehicle was not expecting olives to be deposited. The blue arrows determine the direction of movement along the

311 alley. It is worth mentioning that the path shown in Fig. 5 was kinematically compatible with the
312 kinematic restrictions of the vehicle. Thus, the curvature radius at any point of the path was
313 always larger than $r_{min} = 3m$ (the minimum radius of turning of the service unit, as previously
314 stated).



315

316 Fig. 5: Path planning between olive grove alleys - the path to be tracked by the service unit while harvesting.

317 In order to manage the path to be tracked by the unmanned service unit while harvesting, and
318 taking into account the navigation architecture presented in Fig. 4, the following criteria were
319 used.

320 • Three programs was run in parallel. One per each switch case shown in Fig. 4.
321 • The main parallel program (*Program A*) was in charge of the path tracking controller
322 shown herein, the planning and the RTK-GNSS for position estimation.
323 • The first secondary parallel program (*Program B*) was in charge of the obstacle alert. It
324 continuously searches for an obstacle during the unmanned service unit motion.
325 • The second secondary parallel program (*Program C*) was in charge of the full load alert.
326 If the service unit detected the full load status of the trailer, it re-sampled the path in order

to reach the shelter as soon as possible without violating the kinematic and dynamic constrains of the vehicle.

The algorithms shown in Algorithm 1, 2 and 3 (below) present how the three parallel programs were coordinated. Algorithm 1 shows the execution of *Program A*. As can be seen, Algorithm 1 was in charge of the trajectory controller. In line of code (1), two paths are generated: $Path^1$ -related to the harvesting problem, shown in Fig. 5 -and $Path^2$- related to a full load situation (it is a path generated from the robot's current position to the shelter, according to the information provided by $Path^1$)-. $Path^1$ was followed by the unmanned service unit until a full load flag is detected. In that case, the service unit follows $Path^2$ (*ReferencePath* is an auxiliary array). In line of code (2), $robot_k$ is the instantaneous pose of the service unit within the environment. The switching flag δ was defined in line of code (3); by default, $\delta = 1$. If $\delta = 3$, the service unit has detected the *full load* condition. Then, the reference path is updated to $Path^2$; *Program A* performs $Path^2$ navigation until \$\delta\$ changes its value - *while*-loop, lines of code (7)-(11)-. The command *pause* in lines of code (10), (17) and (22), is a mandatory pause of the robotic system (Δ_t in Eq. (3)). Such pause is used to update the internal values of the system. In this work, $\Delta_t = 0.1s$.

Algorithm 1: Program A: main system

1. Let $Path^1$ and $Path^2$ be two trajectories generated by the system. Let *ReferencePath* the path used by the trajectory tracking controller shown in Fig. 4. Initially, *ReferencePath* = $Path^1$.
2. Let robot_k be the pose of the unmanned car-like service unit within the environment.
3. Let $\delta = 1$ be the switching criterion. If $\delta = 1$, *Program A* is executed; if $\delta = 2$, then *Program B* is executed; if $\delta = 3$, *Program C* is executed.
4. **FOR** k = 0 TO **END** of $Path^1$ **do**
5. IF $\delta = 3$ then

```
6.          ReferencePath = UpdatePath(robot_k, Path^2)
7.          WHILE δ == 3 do
8.          [u_k, ψ_1,k] = ControlCom(robot_k, ReferencePath)
9.          robot_k = RTK-GNSS
10.         pause;
11.       end while
12.       ReferencePath = UpdatePath(robot_k, Path^1)
13.       δ = 1
14.       end if
15.       If δ = 2 then
16.       WHILE δ == 2 do
17.          pause;
18.       end while
19.      endif
20.       [u_k, ψ_1,k] = ControlCom(robot_k, ReferencePath)
21.       robot_k = RTK-GNSS
22.       pause;
23.    end for
```

On the other hand, if $\delta = 2$ {lines of code (15)-(19)}, then an obstacle is detected during the service unit driving whether tracking *Path*^{1} or *Path*^{2}. The vehicle stops until the obstacle is out of view of the laser range scanner, as stated earlier in Section 2.2.1. This behaviour is useful for avoiding collisions between field workers and the service unit. However, if $\delta = 1$, then the service unit follows *Path*^{1}, the harvesting trajectory {lines of code (4)-(23)}. The trajectory controller commands are generated by $ControlCom(robot_k, ReferencePath)$ {lines of code (8) and (20)} which represent Eq. (10), its tangent inverse and Eq. (11). In addition, lines of code (9) and (21), show the service unit's pose estimation using the RTK-GNSS receiver incorporated on the vehicle. Algorithms 2 and 3 show *Program B* and *Program C* respectively. It is worth mentioning that *Program A*, *Program B* and *Program C* all run in parallel, implemented on *OpenMP*, until *Program A* finishes *Path*$^{1.}$

Algorithm 2: Program B: first secondary program.

```
1. Boolean ∝= false
```

```
2.  WHILE (1) do
3.    ∝= ObstacleLaserReadings
4.    if ∝= true then
5.      δ = 2
6.    end if
7.    pause
8.  end while
```

In Algorithm 2, *Obstacle Laser Readings* is a Boolean variable that is true if the laser scan finds an obstacle within the service unit's path. Otherwise, it is false.

As in Algorithm 2, Algorithm 3 shows the case for the *full load* condition, where the Boolean variable α is true if the *full load* situation is observed.

Algorithm 3: Program C: second secondary program.

```
1.  Boolean ∝= false
2.  WHILE (1) do
3.    ∝= FullLoad
4.    if ∝= true then
5.      δ = 3
6.    end if
7.    pause
8.  end while
```

Path planning

The criteria used for managing the trajectories $\Omega = Path^1$ and $\Omega = Path^2$ in Algorithm 1 are based on the harvesting needs of the process but without losing the path shape shown in Fig. 5. Briefly,

- $Path^1$*:* used by the service unit to navigate through alleys performing harvesting aid tasks.
- $Path^2$*:* once the *full load* condition is detected, the vehicle should reach the shelter as soon as possible, unload the olives and return to the point where the full load alert took place to continue the harvesting process. Therefore, once the *full load* condition is detected, and considering that, the vehicle should not move backwards, the system

411 calculates the shortest path between the vehicle and the shelter following the general path
412 shown in Fig. 5. In order to do so, a decision-based algorithm was implemented in order
413 to decide which branch of the path is more appropriate to take (Lavalle 2006). Figure 6
414 shows this case. Once the system detects the *full load* situation, the trajectory from the
415 vehicle's position to the shelter and vice versa is re-sampled incrementing Δ_x and Δ_y -
416 dotted blue line in Fig. 6-, as long as they do not saturate the vehicle's actuators. The
417 solid red dot in Fig. 6 represents the position in which *ReferencePath* in Algorithm 1
418 changes from $Path^1$ to $Path^2$. Once the vehicle returns from the shelter, *ReferencePath* =
419 $Path^1$ and the harvesting process continues.

420 • The system stops the navigation strategy once all alleys have been covered by the
421 unmanned service unit and its trailer.

422 It is worth mentioning that in this research and for safety purposes, the maximum allowed
423 traction speed of the service unit was set to 0.5 m s^{-1}.

424 Yield was expressed in mass of harvested olives and is obtained from the weight increments
425 in the continuous records of the load cell of the trailer. Three loading locations were established
426 in each alley and data from one row can be distinguished from the other (see Fig. 7).

427 To determine the yield of each row section, coordinates of the left and right row sides were
428 averaged and, the olive masses were added. After this process, 78 points were obtained in an
429 approximately 6×25 m grid across the 1.25 ha grove.

430 These data was spatially interpolated on a 3 m grid basis within the boundaries of the plot
431 using an ordinary punctual krigging or Gaussian process regression based on a global variogram.
432 The process is executed using VESPER software (Minasny et al. (2005)).

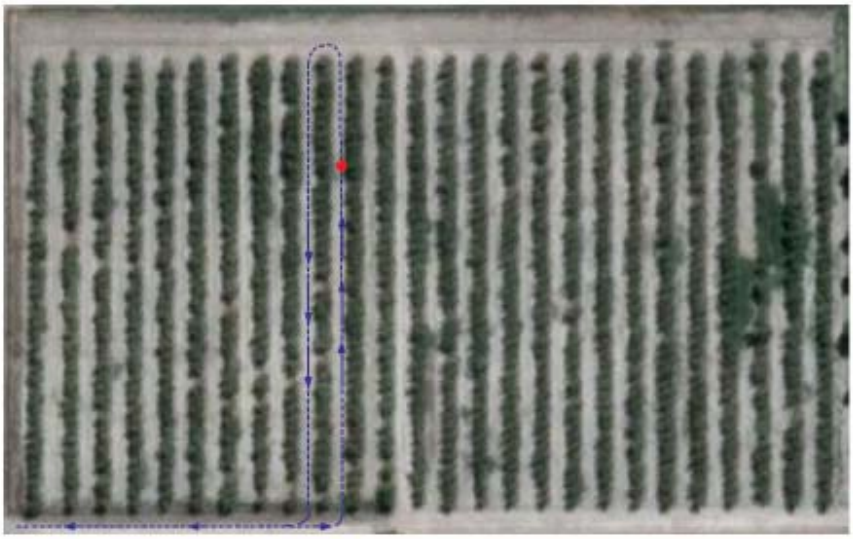


Fig. 6: Trajectory planning with *full load* condition detected. Trajectory planning when full load position is detected (red dot). The service unit plans a trajectory from its current position to the shelter.

A cluster analysis using the k-means algorithm was performed to see whether the yield map can be reclassified into two classes (high and low olive yield). This process is executed with the JMP Pro, Version 11 software (SAS Institute Inc., Cary, USA) as shown in Arnó et al. (2011).

3. Results

3.1 Controller Results

Figure 8 shows the result of implementing the path tracking controller presented in Section *Path Tracking Controller Design*. The distance between any two consecutive points in Ω was set to 0.1 meters. Figure 8 shows the generated trajectory (dotted blue) and the initial position of the service unit $[x_0\ y_0\ \theta_0\ \psi_{1,0}]^T = [5\ 0\ \pi/2\ \ 0.8]^T$; k_θ was set to 0.5 in Eq. (10) and Eq. (11); the top three figures show different snap-shots of the trajectory followed by the robot. The figure at the bottom left shows the error evolution of the controller ($\Delta_x = x_{ref,k} - x_k$) and $\Delta_y = y_{ref,k} -$

y_k); as it can be seen, the errors tend to zero which implies that the service unit (with its trailer) follows the planned trajectory. In addition, the centre right figure shows the control signals of the experiment. The solid black line represents the traction velocity -as it can be seen, the traction velocity does not exceed its limit-; the solid green line represents the angular velocity (w_k) whereas the solid blue line represents the heading ($\psi_{1,k}$) control signal. The heading control command does not exceed its maximum values (set to $\pm 45\,^\circ = \pm 01.05 rad$) , solid red lines.

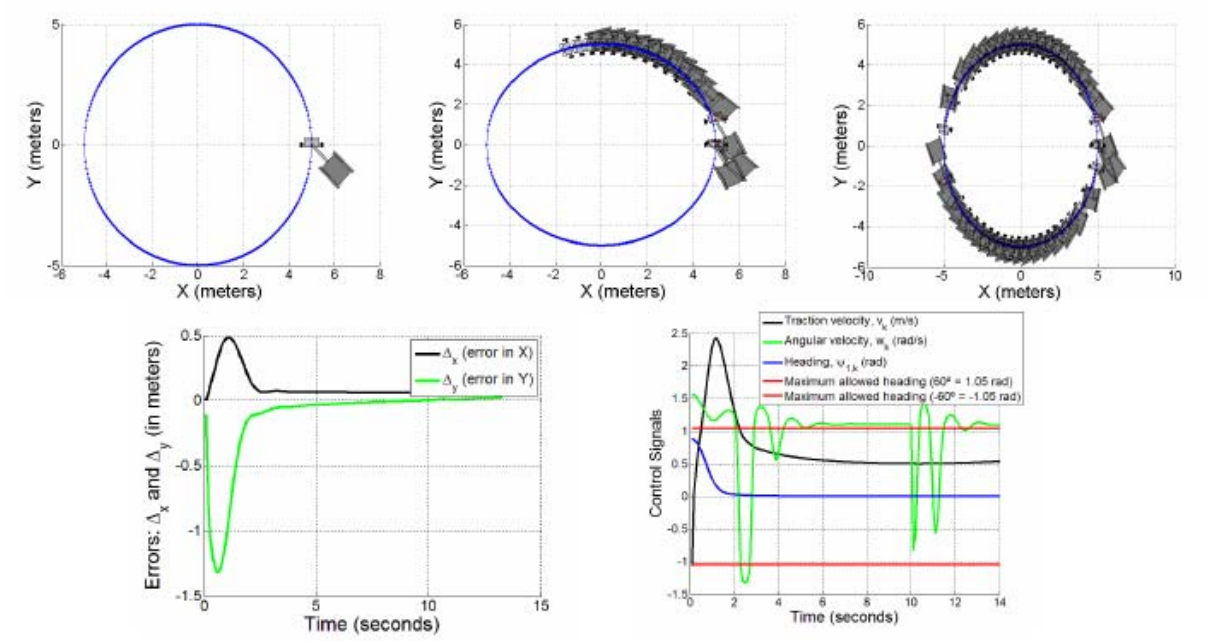
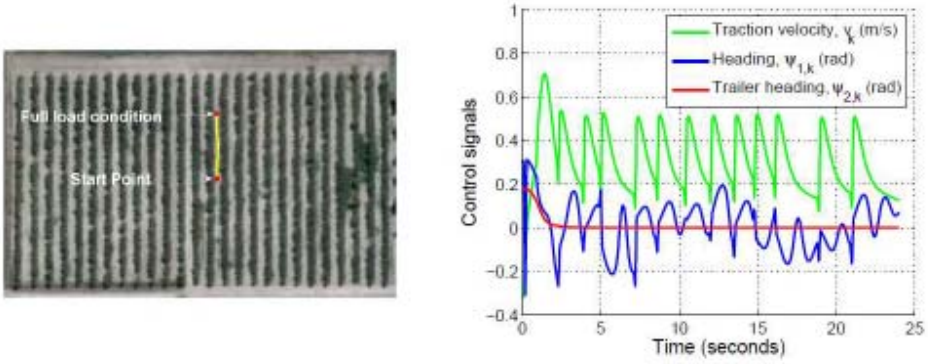


Fig. 8: Simulation results of the path tracking controller. The service unit with a trailer attached to it followed a previously established circular trajectory (dotted blue line). The top three figures show different images from the simulation (initial pose until traversing the entire path). The bottom-left figure shows that the path tracking errors tend to zero ($\Delta_x \rightarrow 0$, $\Delta_y \rightarrow 0$) as the vehicle follows the planned path. The bottom-right figure shows the control command signals over time.

As expected, Fig. 8 shows that, despite of the initial position of the trailer, it tended to follow the orientation of the service unit as it navigated along the planned trajectory. A more complete

464 test of the controller (and the family of controllers associated with this new methodology can be
465 found in Auat Cheein Scaglia (2014)).
466
467 *3.2 Field Results*
468 The experiments were carried out in the olive grove shown in Fig. 3. The following
469 considerations were taken into account:
470 • The path followed by the unmanned service unit and its trailer was the one shown in Fig.
471 5.
472 • The sampling time of the system was set to $\Delta_t = 0.1$ seconds (*pause* in Algorithms. 1, 2
473 and 3) in Eqs. (10) and (11), $k_\theta = 0.5$.
474 • The passive trailer carried up to a maximum 300 kg of olives. Each olive tree can
475 produce (in the best case) 45 kg of olives.
476 • The on-board batteries and fuel of the service unit allowed for ~ 8-h of harvesting work.
477 • The olive grove terrain was not flat, it contained rocks and depressions, but slippage
478 situations were not considered in this research.
479 • The alleys were 6-m wide and the service unit was approximately 2-m wide.
480 Two experimental results are shown herein. Figure 9 shows the case when the service unit
481 navigated to collect olives from the grove (where field workers deposited such olives in the
482 hopper). The figure to the left shows the path tracked by the vehicle from its starting position,
483 solid red dot, until it detected the *full load* condition. The starting position was the point in the
484 environment where the vehicle detected a *full load* condition during the previous stage of the
485 experiment. The path navigated by the service unit is drawn in dotted yellow line. The figure to
486 the right shows the control signal results of the experiment. As it can be seen, the traction

487 velocity remains under 0.5 m s^{-1} as previously stated. In addition, the heading command remains
488 under its saturation values, shown in Fig. 8. Additionally, this figure shows that due to the rocks
489 and the depressions in the terrain, the traction velocity was not as smooth as that shown in Fig. 8.



490

491 Fig. 9: Collecting olives. The first two figures to the left show the path tracked by the unmanned service unit (dotted
492 yellow line) while collecting the olives from the field until a *full load* situation is detected by *Program C*, see
493 Algorithm 3; the figure to the right shows the control signals, the traction velocity (v_k) and the vehicle's heading
494 ($\psi_{1,k}$) imparted to the vehicle and the trailer's heading ($\psi_{2,k}$).

495

496 Figure. 10 shows the case where the service unit has to unload the olives in the shelter and
497 return to the olive grove. In Fig. 10 top left the case when the service unit leaves the *full load*
498 situation point in order to reach the shelter (dotted yellow line) is shown, whereas in the top right
499 of the same figure the vehicle returns from the shelter with the corresponding empty trailer to the
500 previous leaving point within the field in order to continue aiding harvest (dotted yellow line).
501 The bottom left figure of Fig. 10 shows the control command signals obtained during the
502 experiment. The traction velocity remains bounded by the maximum speed and the heading
503 command is also bounded, as previously shown in Fig. 8. The bottom centre figure shows a
504 reconstruction of the olive environment and the full navigation performed by the vehicle. The
505 solid red crosses represent the first and last olive tree from each alley, represented by solid blue

segments. The olive tree position information was previously available from satellite information of the environment. The dotted black line shows the path followed by the vehicle. As it can be seen, the vehicle did not collide with elements within the environment. The bottom right figure is a close-up image of the previous one.

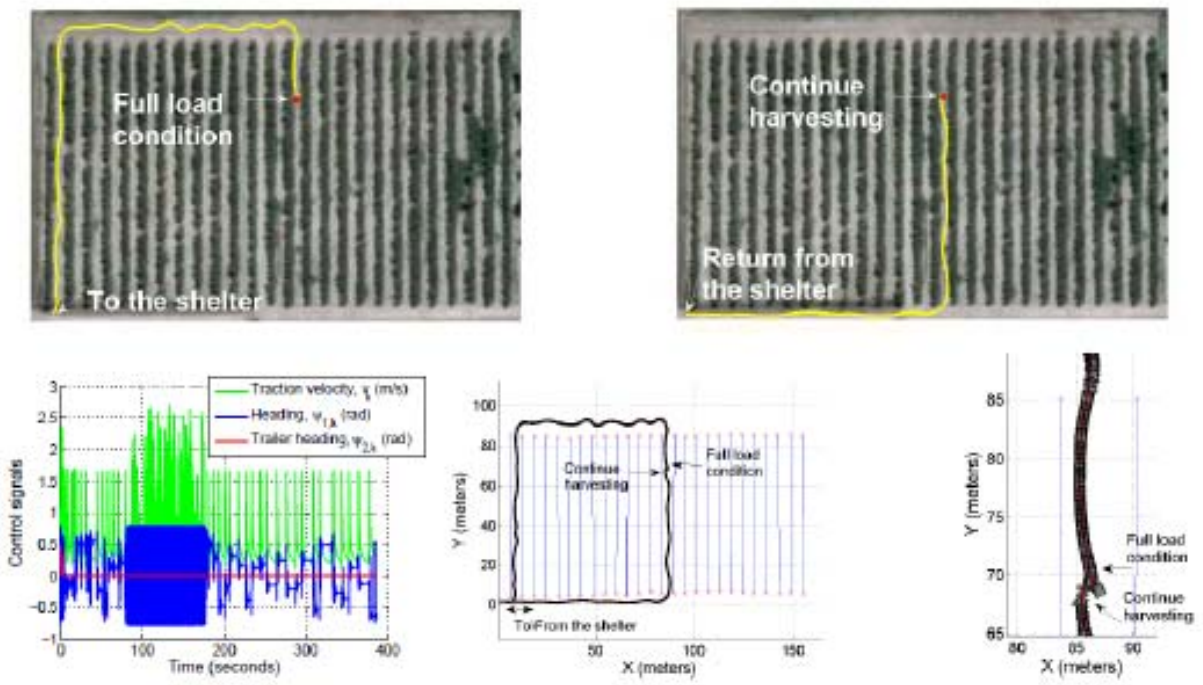


Fig. 10: *Full load* situation. The service unit navigates from its current pose within the grove to the shelter to unload the olives and then return to the grove. The top left shows the case when the trailer detects the *full load* situation and the vehicle navigates up to the shelter in order to unload the olives (dotted yellow line); the top right figure shows the trajectory followed by the vehicle (dotted yellow line) once the trailer is empty and the vehicle has to continue aiding harvesting from the point where it left the grove. The bottom left figure shows the control signals generated during the navigation, the bottom centre and bottom right figures show a reconstruction of the navigation path performed by the vehicle and the alleys from the environment (and its corresponding close-up). As it can be seen, the vehicle does not collide with elements within the environment.

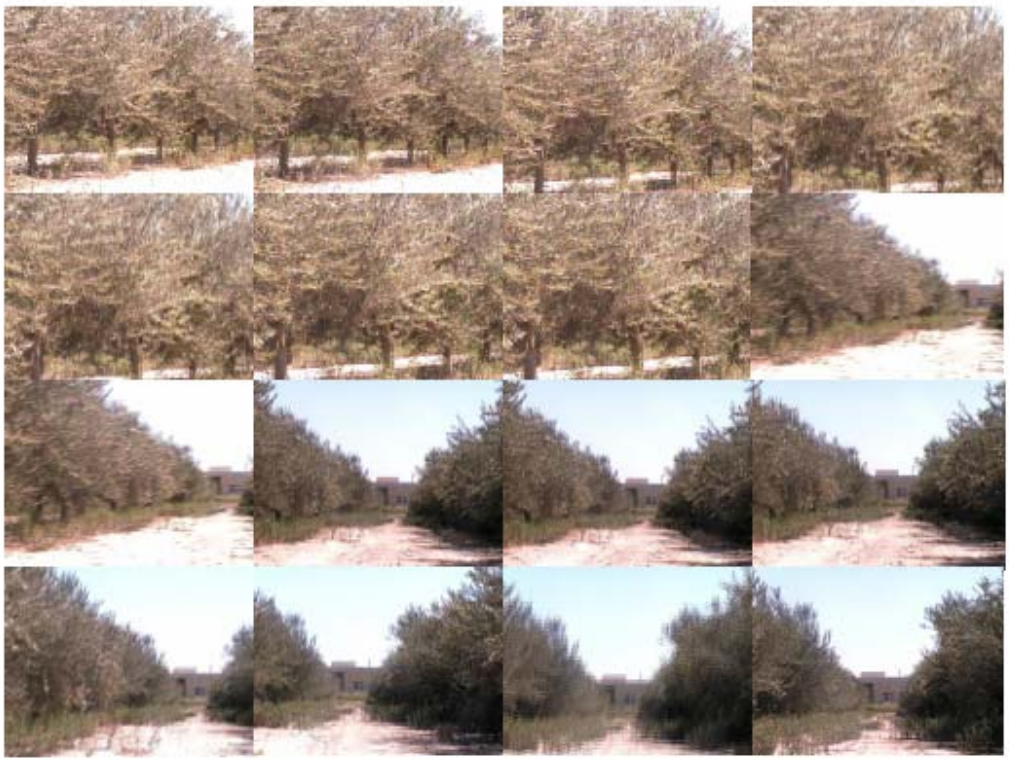


Fig. 11. Different images of the olive grove during the experiments.

Finally, despite the image shown in Fig. 3, during the field experiments the grove was infested with weeds, as shown in the images in Fig. 11. In addition, Fig. 12 shows the path travelled by the service unit (dotted blue line) along a single alley (that shown in Fig. 11). The LiDAR acquired data from the trees and the weeds (green points).

3.3 *Harvesting time*

As previously stated, due to energy resource restrictions, the service unit was able to perform autonomously up to 8 h. However, such time was sufficient for harvesting the grove shown in Fig. 11. In fact, the service unit was able to harvest the entire grove in approximately 7 h. According to field data, once collected, the human transportation of the olives to the shelter can

take up to 13 h if only the transportation time of the olives from their trees to the shelter is considered as being performed by the usual human labour force of 5 field workers ha^{-1}. This represents an improvement of approximately 42% in harvesting time. However, such improvements depend on the number of field workers per hectare.

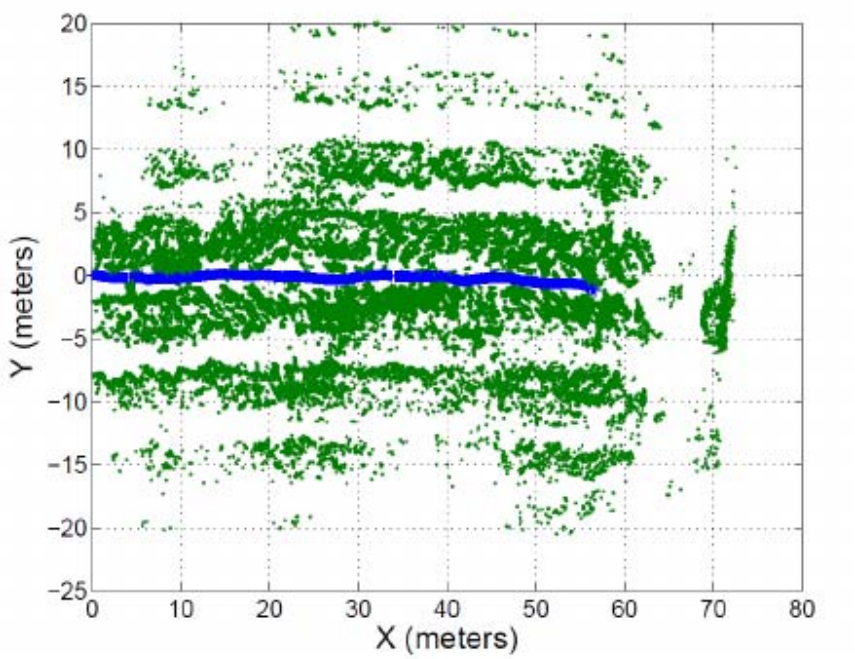


Fig. 12: LiDAR reconstruction of the environment. Reconstruction of an alley from Fig. 11 using LiDAR information (green points). The path travelled by the service unit is shown in dotted blue line.

3.4 *Yield mapping*

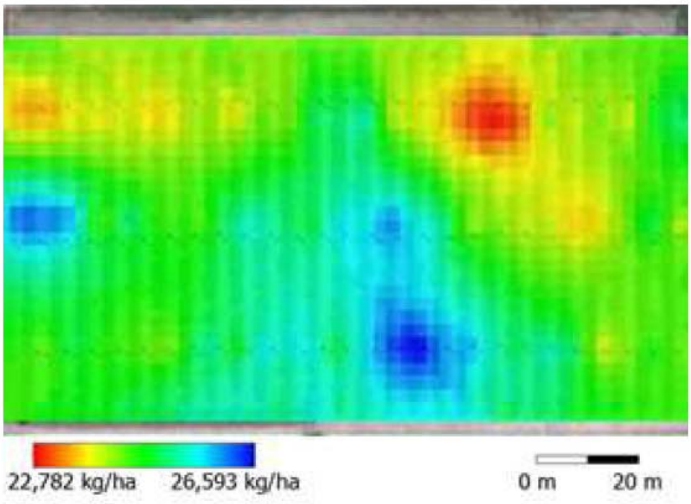
The 78 data points for mass harvested had moderate variability (CV = 10.30%). However, this variability justified a spatial analysis to try to establish whether there is a pattern in the distribution of yield across the grove and whether it can be classified into different classes. An exponential model was adjusted to the variogram and the yield map obtained is shown in Fig.13 a; the interpolated data maintained the same average value but presented a lower variability (CV = 2.62%) and a shorter range. While most of the grove showed an average yield, there were

clearly delimited areas presenting lower (red) and higher (blue) yield values. Figure 13b shows the re-classified map where two significantly different classes have been depicted: low yield (red) and high yield (blue). The classes were structured and compact. The high yield class had an average yield of 25,331.25 kg ha^{-1} and represented a slightly smaller area than the low yield class. The latter has an average yield of 24,243.75 kg ha^{-1}.

4. Discussion

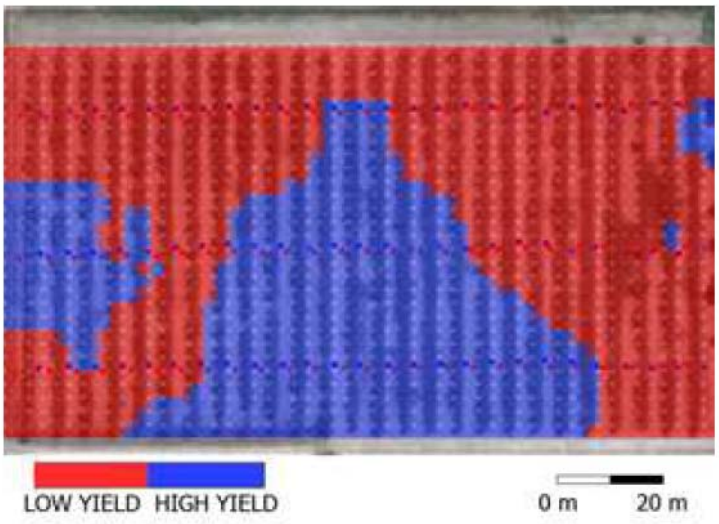
During the experimentation, a number of lessons were learned.

1. The 42 - 45% improvement in the olive harvesting time was calculated according to the basis of 5-7 workers ha^{-1}. However, such improvements could be lower if more field workers were assigned to the harvesting process. Nevertheless, this contradicts the current agricultural situation in Chile, as stated earlier.
2. In the harvesting navigation strategy, once the system detects a *full load* situation, the service unit drives to the shelter to unload the olives. Despite the fact that the experiments showed an improvement of 42 - 45% in the harvesting time better results might be obtained if instead of unloading the olives at the shelter, the full load trailer is left alongside the grove (see Fig. 5) and an empty trailer is attached to the service unit.
3. In the experiments, only one passive trailer was used for harvesting. Considering that the service unit is able to tow up to 300 kg, then harvesting times should capable of further improvement if more than one trailer is used. In this scenario, since trailers are passive, the path tracking control approach shown here would be the same, as well as trajectory reference generation, since no extra restrictions are added into the robot's holonomy. However, the *full load* condition should be updated if an extra trailer is added.



570

571 *(a)*



572

573 *(b)*

574 Fig. 13: Yield maps of the olive grove, (a) shows the yield raster map (b) shows a classification map with
575 two zones - low and high yield.

576 *4.* The olive grove shown in Fig. 11 is an experimental grove. Therefore, slippage and
577 drastic uneven terrain situations were not considered during the experiments.

578 *5.* The layouts of commercial groves are very variable and they also depend on the nature of
579 the crop. This can affect both positively and negatively on the GNSS receiver. Therefore,

using RTK- GNSS as a sole localisation system might cause issues such as the ones mentioned in Auat Cheein & Carelli (2013). Other approaches could be used instead (such simultaneous localisation and mapping- techniques, SLAM). Additionally, the alleys shown in Fig. 12 are 6 m wide, which ensures a free collision driving of the service unit. Better refinement of the path planning technique could be expected for narrower alleys.

6. Although a LiDAR system was used for detecting field workers, a secondary sensor is needed in order to reinforce such crucial task. For instance, this could consist of ultrasonic range sensors or artificial vision systems. Ultrasonic range sensors offer better solutions in poor visibility, and artificial vision systems might help with field worker detection under dusty situations.
7. The measurement and storage of the harvest yields using the trailer's load cell and the corresponding coordinates by GNSS or alternative systems has proved to be a simple but efficient method to subsequently build yield maps. Such yield maps are useful tools to understand the variability of crops as well as for decision making related to more sustainable and efficient use of agricultural inputs (fertilisers, agrochemicals, irrigation, etc.) in the framework of precision agriculture.
8. The yield map could be improved by allowing the operators to load the trailer anywhere along the alleys. This way the spatial resolution of data would increase and the yield map would be more accurate.

However, several open issues arose during the field experimentation that will lead our future research. They are enumerated as follows:

1. The interaction of field workers and machinery (as previously stated in Auat Cheein & Carelli (2013)) when they share the same environment and they complement each other in the agricultural task needs evaluation, particularly from the point of view of field workers.
2. The improvement in harvesting time might be reflected in an improvement in the quality of the olives since as they are collected they could be stored.

5. Conclusions

A path tracking controller for an unmanned articulated car-like type service unit with a passive trailer has been presented. The proposed trajectory controller was asymptotically stable and it can be implemented on a service unit with more than a single trailer without changing its formulation. In addition, given the algebraic nature of the proposed trajectory controller, a re-sampling trajectory approach was also implemented, in order to avoid actuator saturation problems while reaching the service unit's shelter.

The implementation results shown in this work demonstrated the application of the controller in service units for agricultural tasks. In particular, the example of an unmanned service unit navigating within an olive grove while performing harvesting tasks was presented. The proposed navigation strategy used the trajectory tracking controller to perform the harvesting tasks successfully. The navigation strategy was divided into three main stages: navigation while collecting olives, navigation to unload the collected olives and the reaction to obstacles while moving within the field. The first stage was associated with the load of olives into the passive trailer as the service unit navigated among alleys at a low speed. The second stage was associated with the *full load* condition: the trailer reached the maximum amount of olives.

625 Therefore, the service unit re-planned its trajectory in order to reach the shelter as rapidly as
626 possible, where the olives were unload, and then it returned to the harvesting task. The third
627 stage was associated with the protection of the integrity of the vehicle and the field workers: the
628 unmanned service unit stopped its motion if a field worker, or an obstacle, was near the vehicle's
629 local navigation space.

630 Several field experimental results associated with the proposed case study have been
631 presented herein, validating the proposal. Despite of the fact that the terrain used during the
632 experiments presented rocks and depressions and that, during the experiments, the GNSS signal
633 was lost in some parts of the navigation, the trajectory controller and the navigation strategy
634 were sufficiently robust to perform the harvesting task. On the other hand, the availability of the
635 geo-referenced weights of harvested olives in a continuous way by the combined work of a
636 GNSS and the trailer's load cell provides an easy, efficient and reliable way to build yield maps.
637 This feature could be a very relevant added value to the service unit which has been developed
638 allowing the farmer the implementation of precision agriculture techniques to improve the
639 results.

640 The experimental results led to new open issues, such as the interaction of automated machinery
641 with field workers in shared tasks and spaces. This open issue in particular requires additional
642 analysis in order to establish the field worker’s perception of robots and how to optimise such
643 interactions for the improvement of the production. The second open issue could also lead to a
644 new research study, since olives could enter store more rapidly using the service unit than with
645 manual labour. Such improvements in the collecting time could reduce the exposure of olives (or
646 similar fruits) to climatic changes until they are stored. This might be reflected in an
647 improvement in crop quality.

6. Acknowledgements

The authors would like to thank to the Universidad T\'ecnica Federico Santa Maríia (Chile), to FONDECYT grant 1140575 (CONICYT), BASAL grant FB 0008, to the Institute of Automatics, Universidad Nacional de San Juan (Argentina), and to the University of Lleida (Spain), for their support.

Appendix A. Appendix: Convergence Proof of the Path Tracking Controller

The analysis of the zero error convergence of the proposed trajectory controller for the service unit, following the guidelines previously published in Auat Cheein & Scaglia (2014) is presented. Taking into account the discrete kinematic model of the service unit presented in Eq. (3) and the proposed trajectory controller -shown in Eqs. (10) and (11), then $||e_k||_2 \rightarrow 0$ when $k \rightarrow \infty$ if $0 < v_{ref} < \infty$ -the velocity reference-, where $e_k = \left[x_{ref,k} - x_k \;\; y_{ref,k} - y_k\right] = [e_{x,k}\, e_{y,k}]$ is the tracking error with $||e_k||_2 = \sqrt{e_{x,k}^2 + e_{y,k}^2}$. By operating first for the *x*-coordinate of the kinematic model,

$$x_{k+1} = x_k + v_k \cos(\psi_{1,k})\cos(\theta_k)\Delta_t \tag{A.1}$$

with $v_k = \frac{\Delta_x}{\Delta_t}\cos\left(\theta_{ez,k}\right) + \frac{\Delta_y}{\Delta_t}\sin\left(\theta_{ez,k}\right)$. Then, by applying a Taylor series expansion to the cosine in the expression above,

$$\cos(\theta_k) = \cos\left(\theta_{ez,k}\right) - \underbrace{\sin(\theta_{ez,k} + \xi\left(\theta_{ez,k} - \theta_k\right))(\theta_k - \theta_{ez,k})}_{\text{Complementary term, with } 0<\xi<1}$$

and re-ordering the above expression,

$$\cos(\theta_k) = \cos\left(\theta_{ez,k}\right) + \sin(\theta_{ez,k} + \xi\left(\theta_k - \theta_{ez,k}\right))\underbrace{(\theta_{ez,k} - \theta_k)} \tag{A.2}$$

$e_{\theta,k}$

Then, replacing Eq. (A.2) into Eq. (A.1) yields,

$$x_{k+1} = x_k + \underbrace{\left(\frac{\Delta_x}{\Delta_t}\cos(\theta_{ez,k}) + \frac{\Delta_y}{\Delta_t}\sin(\theta_{ez,k})\right)}_{v_k\cos(\psi_{1,k})}$$

$$(\cos(\theta_{ez,k}) + \sin(\theta_{ez,k} + \xi(\theta_k - \theta_{ez,k})))\Delta_t =$$

$$= x_k + \Delta_x cos^2(\theta_{ez,k}) + \Delta_y sin(\theta_{ez,k})\cos(\theta_{ez,k}) +$$

$$\underbrace{+v_k\cos(\psi_{1,k})\cos(\theta_{ez,k}) + \xi(\theta_k - \theta_{ez,k})e_{\theta,k}}_{f}$$

In addition, according to Eq. (5), $\Delta_y = \frac{\sin(\theta_{ez,k})}{\cos(\theta_{ez,k})}\Delta_x$, then,

$$x_{k+1} = x_k + \Delta_x cos^2(\theta_{ez,k}) + \Delta_x sin^2(\theta_{ez,k}) + f_{e_\theta,k} =$$

$$= x_k + \Delta_x + f_{e_\theta,k}$$

and replacing Δ_x in the expression above by Eq. (5) and operating,

$$x_{k+1} = x_k + x_{ref,k+1} - k_x(x_{ref,k} - x_k) - x_k + f_{e_\theta,k} \qquad \text{(A.3)}$$

$$\underbrace{x_{xref,k+1} - x_{k+1}}_{\boldsymbol{e_{x,k+1}}} = k_x \underbrace{(x_{ref,k} - x_k)}_{e_{x,k}} - f_{e_\theta,k}$$

Following the same reasoning applied for the *x*-coordinate to the *y*-coordinate, according to the kinematic model of the car-like service unit and the Taylor series expansion of $\sin(\theta_k)$, the following expression follows:

$$y_{k+1} = y_k + \left(\frac{\Delta_x}{\Delta_t}\cos(\theta_{ez,k}) + \frac{\Delta_y}{\Delta_t}\sin(\theta_{ez,k})\right)$$

$$\underbrace{(\sin(\theta_{ez,k}) - \cos(\theta_{ez,k} + \lambda(\theta_k - \theta_{ez,k}))\underbrace{(\theta_{ez,k} - \theta_k)}_{e_\theta,k})\,\Delta_t}$$

$$sin(\theta_{ez,k})\ expanded\ by\ Taylor\ series, with\ 0 < \lambda < 1$$

678 Then, by operating on the above expression it follows that

$$y_{k+1} = y_k + \Delta_y cos(\theta_{ez,k}) \sin(\theta_{ez,k}) + \Delta_y sin(\theta_{ez,k}) -$$

$$- \underbrace{v_k \cos(\psi_{1,k}) \cos(\theta_{ez,k} + \lambda(\theta_k - \theta_{ez,k}))}_{g} e_{\theta,k}$$

679 According to Eq. (5), $\Delta_x = \frac{\sin(\theta_{ez,k})}{\cos(\theta_{ez,k})} \Delta_y$, hence, by operating the expression above,

$$y_{k+1} = y_k + \Delta_y cos^2(\theta_{ez,k}) + \Delta_y sin^2(\theta_{ez,k}) + g e_{\theta,k} =$$

$$= y_k + y_{ref,k+1} - k_y(y_{ref,k} - y_k) - y_k + g e_{\theta,k} =$$

$$y_{ref,k+1} - k_y(y_{ref,k} - y_k) + g e_{\theta,k}$$

680 Then,

$$\underbrace{y_{ref,k+1} - y_{k+1}}_{e_{y,k+1}} = \underbrace{k_y(y_{ref,k} - y_k)}_{e_{y,k}} + g e_{\theta,k}$$

681

682 Having into account that -see Eq. (3)-,

$$\theta_{k+1} = \theta_k + \frac{\Delta_t}{L_1} v_k \sin(\psi_{1,k})$$

683 and according to Eq. (7),

$$v_k \sin(\psi_{1,k}) = \frac{L_1}{\Delta_t} (\theta_{ez,k+1} - K_\theta(\theta_{ez,k} - \theta_k) - \theta_k =$$

$$= \theta_{ez,k+1} - k_\theta(\theta_{ez,k} - \theta_k)$$

684 Thus,

$$\underbrace{\theta_{ez,k+1} - \theta_{ez,k}}_{e_{\theta,k+1}} = k_\theta \underbrace{(\theta_{ez,k} - \theta_k)}_{e_{\theta,k}} \tag{A.5}$$

685 where $e_{\theta,k}$ is the orientation error. Then, it is possible to see that, in Eq. (A.5), if $0 < k_\theta < 1$

686 then $e_k \to 0$ as $k \to \infty$. In addition, Eqs. (A.3) and (A.4) form the equation system shown in Eq.

$$\begin{bmatrix} e_{x,k+1} \\ e_{y,k+1} \end{bmatrix} = \begin{bmatrix} k_x & 0 \\ 0 & k_x \end{bmatrix} \begin{bmatrix} e_{x,k} \\ e_{y,k} \end{bmatrix} + \begin{bmatrix} -f \\ -g \end{bmatrix} e_{\theta,k} \quad \text{(A.6).}$$

Linear term *Non-Linear term*

Equation (A.6) is a linear system added to a bounded non-linear term. According to Eq. (A.5), the non-linear term of Eq. (A.6) tends to zero as $k \to \infty$. Considering that by hypothesis $v_{ref} < \infty$, then v_k is bounded. Following, it is possible to conclude that Eq. (A.6) is a linear system with a non-linear term that tends to zero as $k \to \infty$. In addition, according to Auat & Scaglia (2014), $||e_k||_2 \to 0$ as $k \to \infty$, which is proof of the proposal.

References

Noguchi N, Will J, Reid J, Zhang Q, (2004), Development of a master-slave robot system for farm operations. Computers and Electronics in Agriculture, vol. 44, pp. 1-19.

Lee W, Slaughter D, Giles D, (1999), Robotic weed control system for tomatoes. Precision Agriculture, vol 1, pp. 95-113.

Malinowski A, Yu H, (2011), Comparison of embedded system design for industrial applications. IEEE Transactions on Industrial Informatics, vol. 7, pp. 244-254.

Perez-Ruiz M, Aguera J, Gil J, Slaughter D, (2011), Optimization of agrochemical application in olive groves based on positioning sensor. Precision Agriculture, vol. 12, pp. 564-575.

707 Wood B, Blair H, Gray D, Kemp P, Kenyon P, Morris S, Sewell A, (2014), Agricultural Science
708 in the Wild: A social network analysis of farme nowledge exchange. Plos One, vol. 9(8):
709 e105203.
710
711 Matsuda E, Hubert J, Ikegami T, (2014), A Robotic Approach to Understanding the Role of the
712 Mechanism of Vicarious Trial-And-Error in a T-Maze Task. Plos One, vol. 9(7): e102708.
713
714 Agricultural Robotics: Unmanned robotic service units in agricultural tasks. IEEE Industrial
715 Electronics Magazine, vol. 7, pp. 48-58.
716
717 Cariou C, Lenain R, Thuilot B, Berducat M (2009), Automatic guidance of a four-wheel-steering
718 mobile robot of accurate field operations. Journal of Field Robotics, vol. 26, pp. 504-518.
719
720 Pota H, Eaton R, Katupitiya J, Pathirana S, (2007), Agricultural robotics: A streamlined
721 approach to realization of autnomous farming. In Proc. of the Second Int. Conf. on
722 Industrial and Information Systems, pp. 85-90, Sri Lanka.
723
724 Rath T, Kawollek M, (2009), Robotic harvesting of gerbera jamesonii based on detection and
725 three-dimensional modeling of cut flower pedicels. Computers and Electronics in
726 Agriculture, vol. 66, pp. 85-92.
727
728 Tanigaki K, Fujiura T, Akase A, Imagawa J, (2008), Cherry-harvesting robot. Computers and
729 Electronics in Agriculture, vol. 63, pp. 65-72.

730

731 Murakami N, Ito A, Will J, Steffen M, Inoue K, Kita K, Miyaura S, (2008), Development of a
732 teleoperation system for agricultural vehicles. Computers and Electronics in Agricultura,
733 vol. 63, pp. 81-88.

734

735 Dong F, Heinemann W, Kasper R, (2011), Development of a row guidance system for an
736 autonomous robot for white asparagus harvesting. Computers and Electronics in
737 Agriculture, vol. 79, pp. 216-225.

738

739 Bochtis, D., Griepentrog, H.W., Vougioukas, S., Busato, P., Berruto, R., Zhou, K. (2015),
740 Route planning for orchard operations. Computers and Electronics in Agriculture, 113, pp. 51-
741 60.

742

743 Anjom, F.K., Rehal, R.S., Fathallah, F.A., Wilken, K.D., Vougioukas, S.G. (2014), Sensor-based
744 stooped work monitoring in robot-aided strawberry harvesting. American Society of
745 Agricultural and Biological Engineers Annual International Meeting 2014, ASABE 2014,
746 7, pp. 5027-5038.

747

748 Bochtis D, Sorensen C, Vougioukas S (2010), Path planning for in-field navigation-aiding of
749 service units. Computers and Electronics in Agriculture, vol. 74, pp. 80-90.

750

751 Mas F, Zhang Q, Reid J, (2008), Stereo vision three-dimensional terrain maps for precision
752 agriculture. Computers and Electronics in Agriculture, vol. 60, pp. 133-143.

Park J, Lee J, (2011), A beacon color code scheduling for the localization of multiple robots. IEEE Transactions on Industrial Informatics, vol. 7, pp. 467-475.

Perez L, Alegre M, Ribeiro A, Guinea D, (2008), An agent of behaviour architecture for unmanned control of a farming vehicle. Computers and Electronics in Agriculture, vol. 60, pp. 39-48.

Rezaee H, Abdollahi F, (2014), A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots. IEEE Transactions on Industrial Electronics, vol. 61, pp. 347-354.

Feng S, Zhang H, (2011), Formation control of wheeled mobile robots based on consensus protocol. IEEE Ing. Conf. on Information and Automation, Shenzen, China, pp. 1-6.

Kim H, Kim B, (2014), Online minimum-energy trajectory planning and control on a straight-line path for the three-wheeled omnidirectional mobile robots. IEEE Transactions on Industrial Electronics, vol. 61, pp. 4771-4779.

Tanaka Y, Murakami T, (2009), A study on straight-line tracking and posture control in electric bicycle. IEEE Transactions on Industrial Electronics, vol. 56, pp. 159-168.

FPGA implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform. IEEE Transactions on Industrial Electronics, vol. 56, pp. 1604-1616.

Lian R, (2011), Intelligent controller for robotic motion control. IEEE Transactions on Industrial Electronics, vol. 58, pp. 5220-5230.

Hwang C, Shih C, (2009), A distributed active-vision network-space approach for the navigation of a car-like wheeled robot. IEEE Transactions on Industrial Electronics, vol 56, pp. 846-855.

Dong W, (2012), Tracking control of multiple-wheeled mobile robots with limited information of a desired trajectory. IEEE Transactions on Robotics, vol. 28, pp. 262-268.

Blazic S (2011), A novel trajectory-tracking control law for wheeled mobile robots. Robotics and Autonomous Systems, vo. 59, pp. 1001-1007.

Martins N, Elyoussef E, Bertol D, De Pieri E, Moreno U, Castelan E, (2011), Trajetory tracking of a nonholonomic mobile robot with kinematic disturbances: A variable structure control design. IEEE Latin America Transactions, vol. 9, pp. 276-283.

796 Coulaud J, Campion G, Bastin G, De Wan M, (2006), Stability analysis of a vision-based control
797 design for an autonomous mobile robot. IEEE Transactions on Robotics, vol. 22, pp. 1062-
798 1069.
799
800 Park B, Yoo S, Park J, Choi Y, (2010), A simple adaptive control approach for trajectory
801 tracking of electrically driven nonholonimic mobile robots. IEEE Transactions on Control
802 Systems Technology, vol. 18, pp. 1199-1206.
803
804 Auat Cheein F, Scaglia G, (2014), Trajectory tracking controller design for unmanned vehicles: a
805 new methodology. Journal of Field Robotics, vol. 31(6), pp. 861-887.
806
807 Nielsen M, Slaughter D, Gliever C, (2012), Vision-Based 3D Peach Tree Reconstruction for
808 Automated Blossom Thinning, IEEE Transactions on Industrial Informatics, vol. 8(1), pp.
809 188-196.
810
811 Zaman Q, Schumann A, (2005), Performance of an Ultrasonic Tree Volume Measurement
812 System in Commercial Citrus Grove. Precision Agriculture, vol. 16, pp. 467-480.
813
814 Aznar F, Pujol F, Pujol M, Rizo R, Pujol M J, (2014), Learning Probabilistic Features for
815 Robotic Navigation Using Laser Sensors. Plos One, vol. 9(11): e112507.
816

817 Alamo S, Ramos Feito M, Precision techniques for improving the management of the olive
818 groves of southern Spain (2012). Spanish Journal of Agricultural Research, vol. 10, pp.
819 583-593.
820
821 Perez-Ruiz M, Slaughter D, Gliever C, Upadyaya S, (2012), Tractor-based real-time kinematic-
822 global positioning system (RTK-GPS) guidance system for geoespatial mapping of row
823 crop transplant. Biosystems Engineering, vol. 111, pp. 64-71.
824
825 Wang Q, Zhang Q, Rovira-Mas F, Tian L, (2011), Stereovision-based lateral offset measurement
826 for vehicle navigation in cultivated subble fields. Biosystems Engineering, vol. 109, pp.
827 258-265.
828
829 Gee C, Bossu J, Jones G, Truchetet F, (2008), Crop/weed discrimination in perspective
830 agronomic images. Computers and Electronics in Agriculture, vol. 60, pp. 49-59.
831
832 Lavalle S, (2006), Planning Algorithms. Cambrdige University Press, Cambridge, MA.
833
834 Minasny B, McBratney A, Whelan B, (2005), Vesper version 1.62. Australian Centre for
835 Precision Agriculture.
836
837 Arnò J, Martinez-Casanovas J, Ribes-Dasi M, Rosell J, (2011), Clustering of grape yield maps to
838 delineate site-specific management zones. Spanish Journal of Agricultural Research, vol. 9,
839 pp. 721-729.